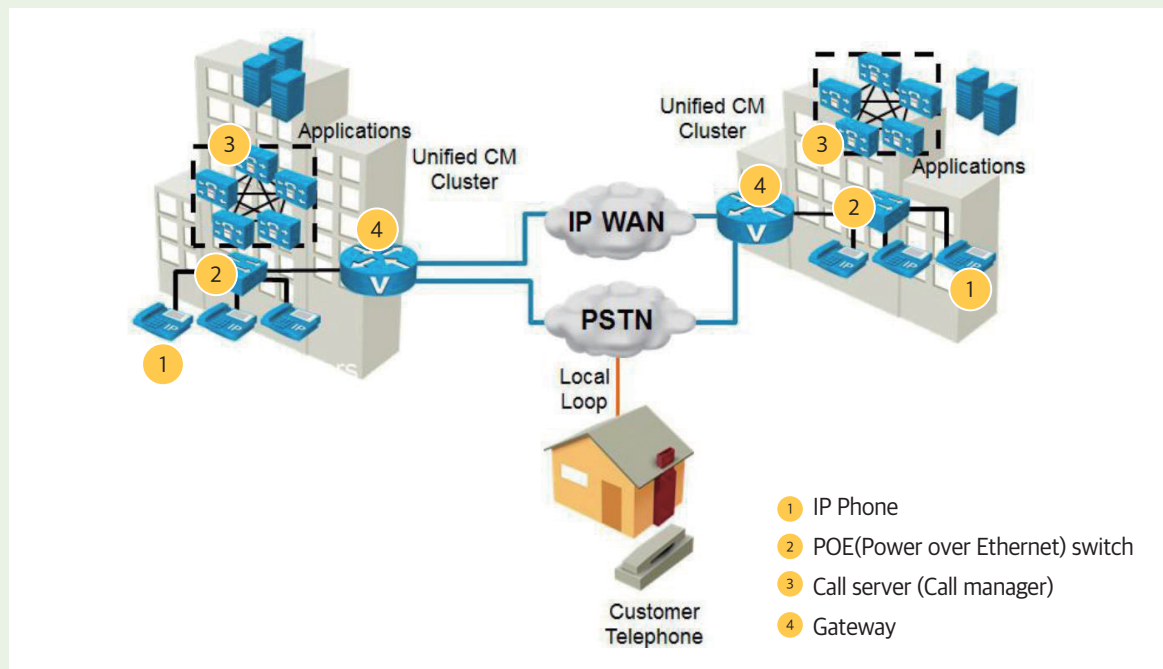


실무 네트워크 Design - 7 : VoIP(Voice over IP)

김해중 KBS 보도기술국

이번 시간에는 IP망을 이용해서 전화를 구현하는 VoIP(Voice over IP)에 대해 설명해드리겠습니다. 일반적으로 VoIP라는 용어보다는, IP telephony로 많이 부르며, 간단히 줄여서 IPT라고 합니다. 사실 VoIP 기술은 많이 복잡하지만, 여기에서는 실무적인 내용만을 간단히 다루도록 하겠습니다. 참고로 국내에 구축된 대부분의 IPT는 avaya, cisco 제품을 사용하고 있으며, 이 글에서는 cisco 기준으로 설명을 하도록 하겠습니다.

IPT 구성요소(4가지)



[그림 1] IPT의 구성 요소들

IP Phone

IP Phone은 전화를 걸 때는, 사람의 목소리(analog) 신호를 Digital 신호인 패킷 형태로 변환하는 A/D converter 역할을 하며, 전화를 수신할 때는 D/A converter 역할을 합니다. IP Phone은 기존의 analog phone에서 제공하지 않는 화상통화, 콘퍼런스의 기능을 제공하지만, 여전히 analog phone에 비해서는 비싸다는 단점이 있습니다.

POE(Power over Ethernet) switch

POE switch는 일반적인 L2 switch의 역할을 하면서, 추가적으로 UTP cable을 통해서 전원도 공급하는 switch입니다. 기존에는 UTP cable을 통해서 data만 전달했다면, 이제는 전원도 같이 공급하는 것입니다. POE 사용으로 IP 전화기에 전원 어댑터가 불필요해서, 배선이 간단해지는 장점이 있습니다. 현재 POE switch는 wireless LAN에 사용되는 무선 AP에 전원을 공급하는 역할로도 많이 사용됩니다.

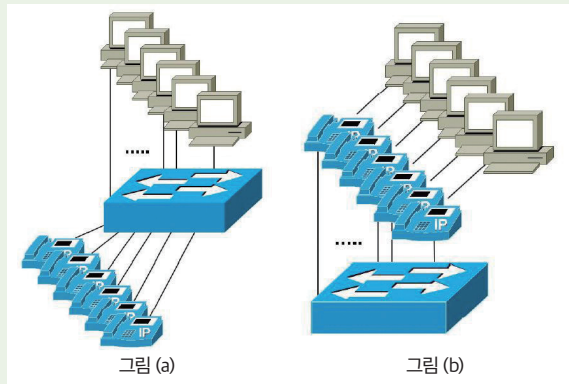
Call server (Call manager)

IPT 환경에서 가장 중요한 장비로, 실질적인 call processing(call 요청, 등록, 해지) 등을 모두 처리합니다. 그리고 call routing table을 가지고, 전화번호를 체계적으로 관리합니다. 참고로 cisco에서는 call server를 call manager(CM)로 부릅니다.

Gateway

IPT 환경에서 기존의 PSTN(기존 전화망)과 연동하기 위해서 사용하는 장비가 gateway입니다. 일반적으로 router에 PSTN을 지원하는 모듈을 추가해서 gateway 역할을 하게 됩니다.

Voice VLAN

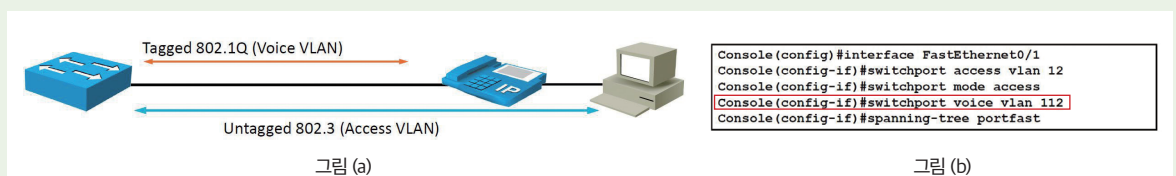


[그림 2] switch에 IP 전화를 연결하는 2가지 방식

기존에는 switch에 PC만 연결되었지만, 이제는 switch에 전화기와 PC가 같이 연결되겠죠? 연결 방법에는 2가지가 있습니다. 첫 번째는 [그림 2-(a)]처럼 전화기와 PC를 switch의 각 Port에 연결하는 방식입니다. 이렇게 되면 switch에 많은 port가 필요하게 되며, switch port가 부족할 때는 추가적인 switch가 필요합니다. 두 번째는 [그림 2-(b)]처럼 switch port 수를 절약하기 위해, 1개의 물리적인 UTP cable을 공유해서 전화기와 PC를 연결하는 방식입니다. 즉 먼저 switch에서 IP 전화기로 연결하며, 다시 IP 전화기에서 PC로 연결하는 방식이며, 일반적으로 이 방식을 선호합니다. 그래서 지금부터는 2번째 방식만을 가정하고 설명하겠습니다.

switch 입장에서 보면, 1가닥의 UTP cable을 공유하다 보니, 자신에게 들어오는 traffic이 voice traffic인지, data traffic인지 구분을 하는데 어려움이 생깁니다. 이것을 해결하기 위하여 voice traffic에는 별도의 vlan을 할당하게 되는데 이것을 voice vlan이라고 합니다.

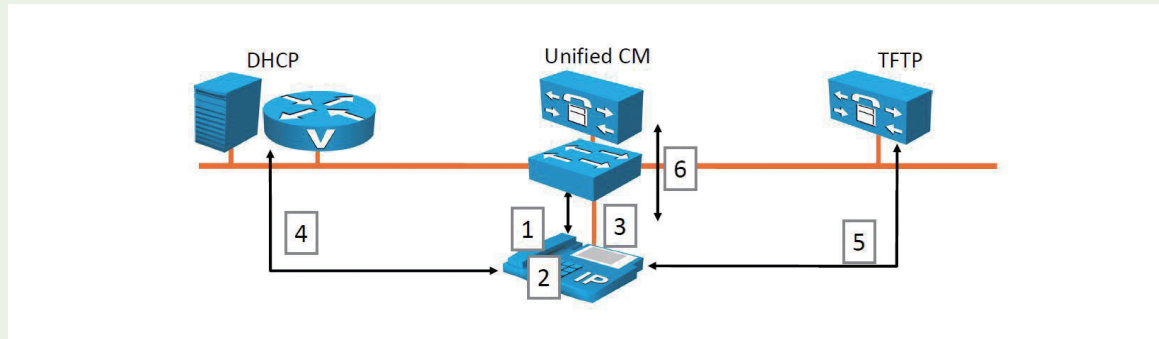
최초 전화기에 전원이 들어오면, 전화기는 자신이 연결된 switch와 CDP/LDDP를 사용하여 voice traffic 전송 시 사용할 vlan을 받아 옵니다. voice traffic을 전송할 때는 받아온 vlan tag를 붙여서 보내게 됩니다. [그림 3-(a)]처럼 switch 입장에서 보면, data traffic은 vlan tag가 없으며(untagged), voice traffic은 vlan tag가 있으니 쉽게 구분이 됩니다. 실제 아주 장황하게 설명을 했지만, 실제 cisco switch에 명령어를 입력할 때는 [그림 3-(b)]에 빨간색으로 표시된 1줄만 추가해주면 되는 아주 간단한 작업입니다.



[그림 3] voice vlan 개념

IP Phone startup process

최초에 IP 전화기는 [그림 4]와 같이 동작을 하며, 설명은 [표 1]을 참조하시면 됩니다.



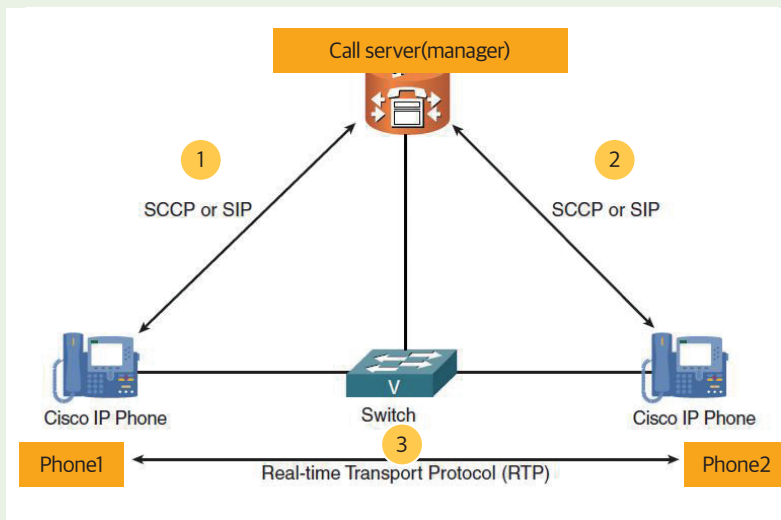
[그림 4] IP Phone startup process

순서	내용
1	IP Phone은 power adapter나 POE switch를 통해 전원을 공급받습니다.
2	IP Phone은 자신의 flash memory 저장되어 있는 image를 불러옵니다.
3	IP Phone은 switch로부터 voice vlan 번호를 할당받습니다.
4	일반적으로 IP Phone은 고정 IP방식을 잘 사용하지 않고, DHCP 서버로부터 동적으로 IP를 할당받는 방식을 선호합니다. DHCP서버는 IP를 할당할 때, IP phone에게 세팅 정보를 제공하는 tftp서버정보도 같이 알려줍니다.
5	IP phone은 tftp서버에 접속해서 세팅 정보를 다운로드 합니다. 그리고 자신이 가지고 있는 firmware와 tftp서버로부터 받은 것을 비교해서, 같으면 다음 단계로 넘어가고, 다르면 tftp서버로부터 firmware를 다운로드 한 후에 리부팅을 합니다.
6	이제 IP phone은 call-server에 접근해서, call-server에 자신을 등록합니다. 그러면 call-server는 IP Phone이 사용할 전화번호를 할당합니다.

[표 1] IP Phone startup process

Call Flow

그림 실제 전화 통화가 어떻게 되는지 [그림 5]를 통해 알아보까요?



[그림 5] Call-flow

최초 사용자가 phone 1의 수화기를 들고 phone 2의 전화번호를 누르면 signal protocol(SCCP, SIP)를 통해서 call server에게 전달합니다. signal protocol은 call server와 IP 전화기 간에 상태정보를 주고받을 때 사용하는 protocol입니다. 최근에 phone에서 가장 많이 사용하는 signal protocol은 SCCP와 SIP입니다. SCCP는 cisco phone 전용이며, SIP는 표준 기술입니다. 점차 SIP phone을 사용하는 추세로 전환되고 있습니다.

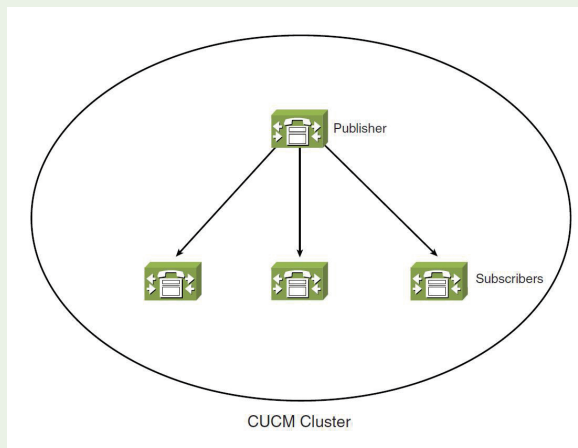
call server는 자신에게 도착한 전화번호를 확인한 후에, 목적지가 phone 2임을 확인합니다. call server는 이제 signal protocol을 사용하여 phone 2의 전화번호를 올리게 해서, phone 1과 phone 2를 연결합니다. call server는 중재하는 사람처럼, 연결해주고 나면 더 이상 call-flow에 간섭하지 않고 빠집니다. 즉 phone 1과 phone 2가 음성 통화를 할 때는, 더 이상 call server를 거치지 않고, 직접 RTP(Real-time Transport Protocol)를 사용하여 phone 간에 직접 통신을 하게 됩니다.

call server가 RTP 통신에 참여 안 하는 이유로는 2가지가 있는데, 첫 번째는 RTP 연결 후에, 혹시 call server에 장애가 발생하더라도, 통화 중인 2대의 phone은 계속 통화를 유지할 수 있기 때문이며, 두 번째는 모든 voice data가 call server를 통해서 가게 된다면, call server가 bottleneck point가 될 수 있기 때문입니다.

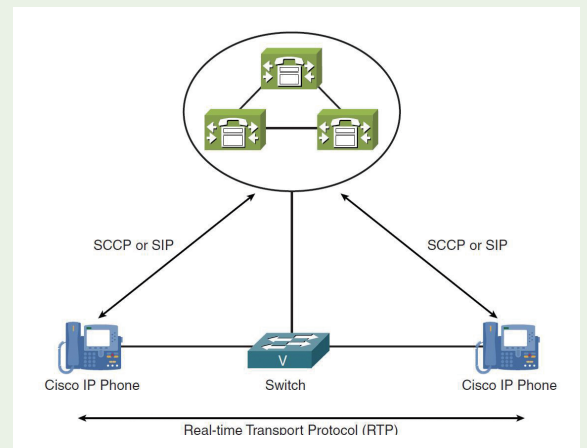
Call server 장애 대비

IP Phone은 최초에 모든 통신을 call server와 합니다. 만약 1대의 call server를 사용하고 있을 때 장애가 발생한다면, 모든 통신이 두절됩니다. 그래서 장애 대비책으로 여러 대의 call server를 사용합니다. 그리고 1대의 call server가 처리할 수 있는 call processing의 한계가 있기에, 여러 대의 call server를 이용해 load-balancing을 구현합니다.

cisco에서는 여러 대의 call server의 집합을 cluster라고 표현을 하며, 그중 1대의 call server가 call database를 메인으로 관리하는데, publisher라고 부릅니다. 나머지 call server는 publisher가 가지는 database의 복제를 받게 되는 구조이며, 이때 이들을 subscriber라고 합니다. cluster를 구성한 후에, 각 ip phone은 3대의 call server를 primary(1순위), secondary(2순위), tertiary(3순위)로 지정해서, 장애를 대비할 수 있습니다. 즉 primary server가 down되면 secondary를 사용하고, secondary server가 down되면, tertiary를 사용하는 방식을 사용합니다. 각 phone마다 primary, secondary, tertiary를 다르게 한다면 로드 밸런싱을 쉽게 구현할 수 있습니다.



[그림 6] 여러 대의 call server의 집합을 cluster라고 하며, publisher가 main database를 가지고 있습니다

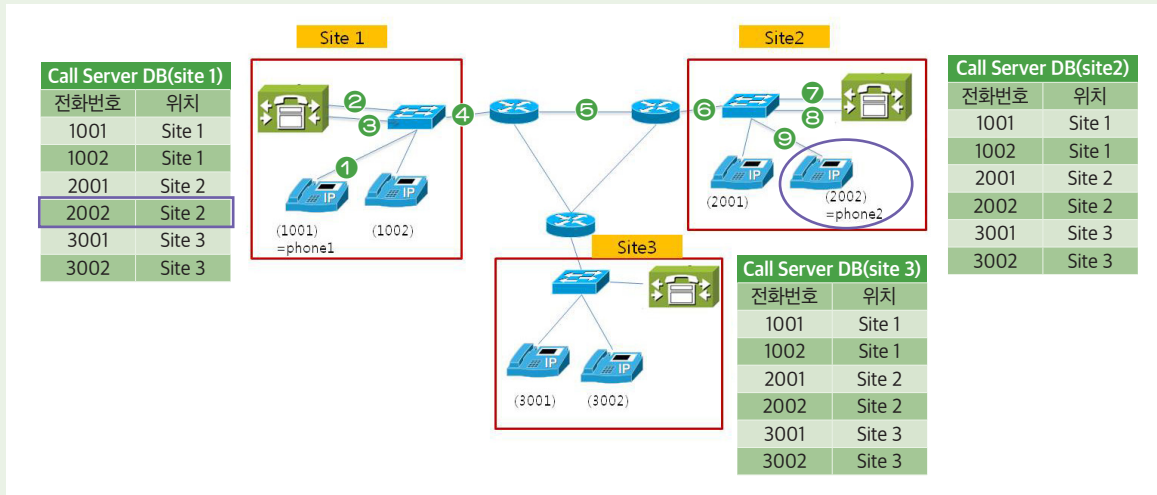


[그림 7] cluster를 사용해 call processing에 대한 로드 밸런싱을 구현할 수 있습니다

Call routing(서로 다른 site 간의 통신)

IP 통신에서 서로 다른 network 간에 통신하려면 routing이 필요하듯이, voice call에서도 서로 다른 전화번호를 가지는 site 간에 전화 통화를 하려면 call routing이 필요합니다. 'ip routing'은 router가 패킷을 수신한 후에, 목적지의 IP 주소를 보고, 목적지로 가는 최적의 경로를 찾아주는 것을 의미하며, call routing은 call server가 목적지 전화번호를 수신한 후에, 자신의 전화번호 database에서 목적지 전화번호까지 최적 경로를 찾아주는 것입니다.

[그림 8]을 통해 phone 1과 phone 2가 어떻게 통신되는지를 살펴보도록 하겠습니다.



[그림 8] call routing은 목적지 전화번호를 보고, 목적지까지의 최적의 경로를 찾아 주는 것을 의미한다

최초 site 1에 있는 phone1에서 site 2에 있는 phone 2의 전화번호인 2002를 누르면, site 1의 call server에 목적지 전화번호 (2002)가 도착합니다. call server는 자신의 전화번호 database를 확인해, 2002가 site 2에 있다는 것을 알게 됩니다. 그러면 site 1의 call server는 site 2의 call server에게 “우리 site의 phone 1이, 너희 site의 phone 2랑 통화하고 싶어 해”라는 메시지를 전달합니다. 메시지를 수신한 site 2의 call server는 해당 정보를 phone 2에 전달해서 phone 2의 전화벨이 울리게 됩니다. phone 2의 사용자가 수화기를 들게 되면, 드디어 phone 1과 phone 2는 RTP로 전화통신을 하게 됩니다.

Call Server DB(site 1)	
전화번호	위치
1001	Site 1
1002	Site 1
2001	Site 2
2002	Site 2
3001	Site 3
3002	Site 3

같은 site에 있는 전화번호
- 자동으로 만들어진다.

다른 site에 있는 전화번호
- 수동으로 만들어야 한다.

[그림 9] site 1의 call server에 존재하는 전화번호 database

그러면 call server가 전화번호 database를 어떻게 만드는지 알아보도록 하겠습니다. 최초에 call server는 자신의 site에 있는 전화번호만 알고 있습니다. 그럼 다른 site의 전화번호는 어떻게 알 수 있을까요? 안타깝게도 자동으로는 알 수가 없습니다. 그래서 관리자가 수동으로 모든 전화번호 database를 만들어야 합니다. 즉 ip routing에서 static routing과 100% 동일하게 모두 수작업으로 해야 합니다. 만약 site별로 전화번호가 수백 개 된다면, 동일한 작업을 수백 번 해야 할 뿐만 아니라, 전화번호 database가 너무 커지게 된다는 문제점이 발생합니다.

그래서 대응책으로 복잡한 것을 간결하게 표현할 수 있는 regular expression(regex)을 사용하게 됩니다. 실제로 regex는 많은 프로그래밍 언어에 사용되고 있으며, 특히 routing protocol인 BGP에서 많이 사용됩니다. [표 2]에는 voice에서 자주 사용하는 regex의 wild card를 정리했으며, [표 3]에는 wildcard를 사용하는 패턴별 예제를 정리했습니다.

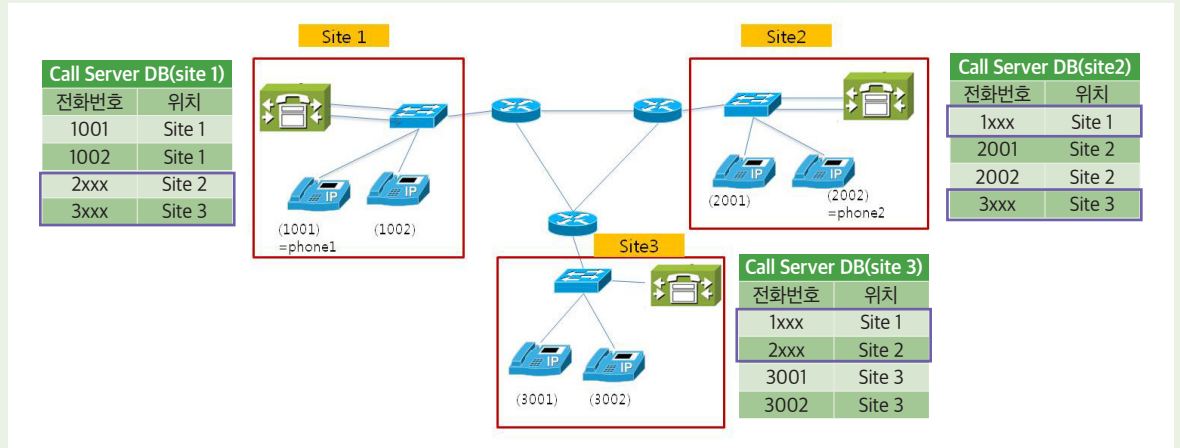
Wildcard	특징	Pattern	결과
[1-9]	1~9중에 임의의 숫자 1개	1234	1234와 정확히 match
[*3-5]	1~9중에서 3~5를 제외한 임의의 숫자 1개	12xx	1200~1299 범위 안에 match
x	단일 digit(0-9, *,#)	1xxx	1000~1999 범위 안에 match
!	Any digit(1개 이상)	13[25-8]6	1326,1356,1366,1376,1386 중 하나와 match

[표 2] VoIP에서 자주 사용하는 regular expression(regex)

[표 3] wildcard를 사용한 패턴별 예제

비록 [그림 10]에는 site마다 비록 전화기가 2대밖에 없지만, 실제로는 수백 대의 전화기가 존재한다고 가정하고 regex로 표현을 하겠습니다. site 1의 입장에서 바라볼 때, site 2의 전화번호는 2001에서 2999사이의 전화번호를 가지고 있으니, 간단히 (2xxx)로

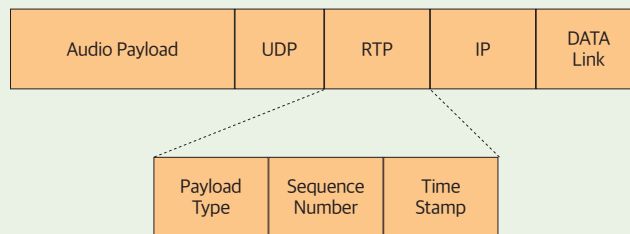
표현할 수 있으며, site 3의 전화번호는, 3001에서 3999사의 전화번호를 가지고 있으니 (3xxx)로 표현할 수 있습니다. 각 site에서 regex를 사용해서 database를 간단히 표현한 것을 [그림 10]으로 확인할 수 있습니다. [그림 8]의 전화번호 database와 비교해 보시면 간결해진 것을 확인할 수 있습니다.



[그림 10] regex를 활용으로 간단히 표현된 database

RTP(Real time transport Protocol)

IP 네트워크에서 audio, video와 같은 실시간 속성을 갖는 트래픽은 UDP로 전송을 합니다. 왜 TCP로 전송하지 않을까요? TCP는 수신단이 패킷을 받으면 ack를 통해서 송신단에게 “나 잘 받았어, 다음 패킷 보내줘” 이렇게 알리게 됩니다. 만약 ack가 일정시간동안 도착하지 않으면, 송신단은 해당 패킷을 재전송해서 안정적인 전송을 보장합니다. 하지만 전화통화 같은 경우, 도중에 패킷이 유실되었다고 해도 재전송을 통해 유실된 부분을 다시 들려줄 수 없습니다. 상대방이 듣지 못했다 하더라도, 전화통화는 실시간으로 계속 진행되어야 하기 때문입니다. 그래서 UDP를 사용하는 것이 더욱 적합합니다. 실제로 audio, video traffic을 전송할 때는 UDP의 단점을 보완하기 위해, RTP를 같이 사용하며, RTP는 크게 3가지 기능을 제공합니다.



RTP Header Information

[그림 11] voice packet 구조

첫 번째로 RTP는 패킷 재정렬 기능을 제공합니다. 송신단의 RTP는 패킷을 전송할 때, 각 패킷마다 번호(sequence number)를 붙입니다. 수신단의 RTP는 도착한 패킷의 sequence number를 확인 후에, 패킷이 순서대로 도착하지 않았을 때는 sequence number 순서대로 재정렬을 합니다.

두 번째로 RTP를 사용해 효율적으로 jitter를 극복할 수 있습니다. jitter란 패킷이 목적지에 도착은 하지만, 각 패킷의 도착시간이 일정하지 않을 때를 의미합니다. 보이스 패킷이 일정하게 도착하지 않으면 음성은 뚝뚝 끊기듯이 재생될 것입니다. 그래서 jitter에 대한 해결책으로 수신단은 Dejitter buffer를 사용합니다. 수신되는 audio를 바로 재생하지 않고, 일정 시간동안 버퍼에 담아두었다가, 버퍼가 다 차면 재생을 시작하게 됩니다. 실시간 재생 프로그램들에서 “버퍼링0%”이런 메시지가 보이는데, 이때 buffer가 Dejitter

buffer입니다. Dejitte buffer의 사용으로 좀 더 지연이 된다는 문제점은 있지만, jitter에 대한 가장 효과적인 대응책입니다. 만약 Dejitte buffer의 크기가 너무 크다면 buffer가 비워지는 것은 막을 수는 있지만, 지연시간이 증가할 것이며, Dejitte buffer의 크기가 너무 작다면 지연시간은 줄일 수 있지만, 버퍼가 쉽게 비워질 위험성이 존재해서 Dejitte buffer를 가변적으로 운영하는 것이 유리합니다. 그래서 송신측의 RTP는 패킷이 만들어 질 때마다 생성된 시간을 timestamp field에 기록해 전송합니다. 수신측의 RTP는 도착한 패킷의 timestamp를 비교해서 지연시간을 알아내며, 이 값을 반영하여 jitter buffer의 크기를 동적으로 조절하여 효율적으로 jitter를 방지합니다.

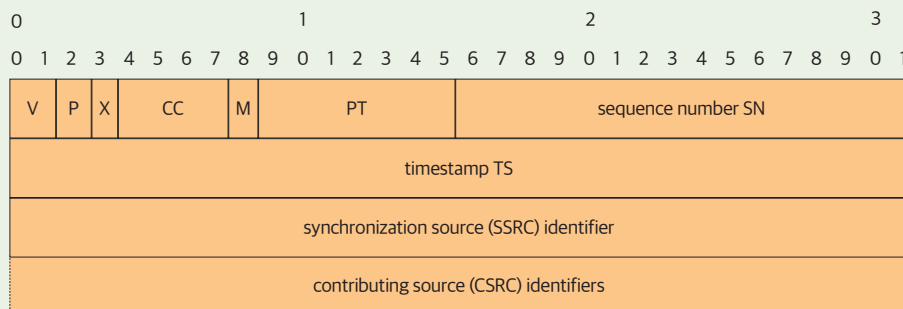
세 번째로 RTP는 silence suppression 기능을 제공합니다. 전화 통화할 때 대화가 순간 단절된 시간에는 패킷을 전송하지 않는 기능을 silence suppression이라고 합니다. 이 기능을 통해 대역폭을 절약할 수 있습니다.

그럼 RTP는 어떤 audio, video 신호를 전송할 수 있을까요? [표 4]에 표현된 다양한 video, audio 신호를 전송할 수 있습니다. RTP는 자신이 전송하는 신호의 내용에 대해, RTP header의 PT(payload type)에 표시합니다. 참고로 최근에 이슈가 되고 있는 IP망을 통한 비압축 SDI 신호 전송(SMPTE 2022-5,6)은 payload type을 98, 99번을 사용하고 있습니다.

PT	Payload	PT	Payload
0	PCMU audio	15	G728 audio
1	1016 audio	18	G729 audio
2	G721 audio	23	RGB 8 video
3	GSM audio	24	HDCC video
4	G723 audio	25	Cel8 video
5	DV14 audio(8khz)	26	JPEG video
6	DV14 audio(16khz)	27	CUSM video
7	LPC audio	28	nv video
8	PCMA audio	29	PicW video
9	G722 audio	30	CPV video
10	L16 audio(stereo)	31	H261video
11	L16 audio(mono)	32	MPV video
12	TPS 0 audio	33	MP2 Tvideo
13	VSC audio	34	H263 video
14	MPA audio	96-127	dynamic

[표 4] RTP가 전송할 수 있는 video, audio 신호

RTP의 헤더구조는 [그림 13]을 통해 확인할 수 있으며, header의 각 field의 특징은 [표 5]에 정리되어 있습니다.



[표 5] RTP-header 구조

V(version)	현재 version은 2입니다.
P(Padding)	0은 padding이 없다는 것을 표시하고, 1은 padding이 존재함을 표시합니다. - RTP에 option header가 존재할 때, option header의 크기가 4byte의 정수배가 되지 않을 경우, 나머지 공간을 0으로 채우는 것을 padding이라 합니다.
X(extension)	option field가 존재함을 알려줍니다.
CC(CSRC Count)	option field인 CSRC의 계수를 의미합니다.
M(Marker)	Silence suppression이 끝나고 말을 시작하는 첫 번째 패킷에 표시합니다.
PT(Payload type)	payload에 어떤 정보를 포함하고 있는지 알려 준다.
sequence number	각 패킷에 번호를 붙이는데 이것을 sequence number라고 하며, 수신단에서 패킷 재정렬 용도로 사용합니다.
Timestamp	패킷의 발생시간을 의미하며, jitter에 대한 대응책으로 사용됩니다.
SSRC	RTP 패킷을 만든 소스를 표시하는데 사용됩니다. -SSRC ID는 각 RTP 세션마다 랜덤한 번호로 구현되며, 당연히 세션에서 고유한 값입니다.
CSRC	CSRC는 conference call을 할 때만 사용되는 option header입니다. - conference를 할 때, 각 사용자의 codec이 다를 때는, 먼저 transcoder에 모든 패킷을 보냅니다. transcoder는 transcoding이 끝난 후에, 패킷을 목적지에 보낼 때, 최초로 이 패킷을 보낸 최초의 송신자의 SSRC를 CSRC에 기록해 보냅니다.

【표 5】RTP header의 각 field의 특징 정리

요약

용어	개념
voice vlan	voice traffic을 전송할 때 사용하는 vlan입니다.
POE	Power over ethernet의 약자로, UTP cable을 통해 data 뿐만 아니라 전원도 같이 공급하는 것을 의미합니다.
cluster	여러 대의 call server의 집합을 cluster라고 합니다. cluster 구성을 통해 call processing에 대한 로드 밸런싱을 구현할 수 있습니다.
RTP	UDP로 video, audio 신호를 전송할 때, UDP의 단점을 보완하기 위해 같이 사용합니다. RTP는 패킷 재정렬, 자동 jitter 버퍼 크기 조절, silence suppression을 제공합니다.
silence suppression	전화통화 중에 침묵할 때는 패킷을 보내지 않는 기능입니다.
call routing	call server가 목적지 전화번호를 수신한 후, 자신의 전화번호 database에서 목적지 전화번호까지 최적 경로를 찾아주는 것입니다.

과거에 VoIP는 전화 통신의 개념이 강했지만, 현재의 VoIP는 더 이상 voice 신호만이 아닌, video 신호까지도 IP망을 통해 전달하는 Video over IP 개념으로 바뀌고 있습니다. 즉 전화기만 IP를 이용하는 것이 아니라, 전통적인 video 장비들인 Camera, Video Mixer들도 점차 IP 기반으로 video 신호를 전송하게 될 것입니다.

현재 많은 vender들은 IP망을 통한 video 신호에 대한 다양한 솔루션을 제공하고 있습니다. 방송기술 엔지니어에게 조금씩 조금씩 IP 기술이 다가움을 느끼게 됩니다. 📺